

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Number	09/883,301	Confirmation No.:	9724
Applicant	Jeffrey A. Bedell et al.		
Filed	June 19, 2001		
Title	METHOD AND SYSTEM FOR IMPLEMENTING DATABASE CONNECTION MAPPING FOR REPORTING SYSTEMS		
TC/Art Unit	2141		
Examiner:	Chirag Patel		
Docket No.	53470.003029		
Customer No.	<b>21967</b>		

**APPEAL BRIEF**

## TABLE OF CONTENTS

	Page
<b>I. REAL PARTY IN INTEREST .....</b>	<b>1</b>
<b>II. RELATED APPEALS AND INTERFERENCES.....</b>	<b>1</b>
<b>III. STATUS OF CLAIMS.....</b>	<b>1</b>
<b>IV. STATUS OF AMENDMENTS .....</b>	<b>2</b>
<b>V. SUMMARY OF CLAIMED SUBJECT MATTER .....</b>	<b>2</b>
<b>A. Concise explanation of Independent Claim 1</b>	<b>2</b>
<b>B. Concise explanation of Independent Claim 7</b>	<b>3</b>
<b>C. Concise explanation of Independent Claim 13</b>	<b>3</b>
<b>VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL.....</b>	<b>4</b>
<b>VII. ARGUMENT .....</b>	<b>4</b>
<b>VIII. CONCLUSION.....</b>	<b>19</b>
<b>IX. CLAIMS APPENDIX .....</b>	<b>20</b>
<b>X. EVIDENCE APPENDIX .....</b>	<b>23</b>
<b>XI. RELATED PROCEEDINGS APPENDIX.....</b>	<b>24</b>

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Number	09/883,301	Confirmation No.:	9724
Applicant	Jeffrey A. Bedell et al.		
Filed	June 19, 2001		
Title	METHOD AND SYSTEM FOR IMPLEMENTING DATABASE CONNECTION MAPPING FOR REPORTING SYSTEMS		
TC/Art Unit	2141		
Examiner:	Chirag Patel		
Docket No.	53470.003029		
Customer No.	<b>21967</b>		

**APPEAL BRIEF**

In response to the Office Action dated October 4, 2005, finally rejecting pending claims 1-18, Appellants respectfully request that the Board of Patent Appeals and Interferences reconsider and withdraw the rejections of record, and allow the pending claims, which are attached hereto as an Appendix.

**I. Real Party In Interest**

The real party in interest is Microstrategy, Incorporated as assignee of the entire interest in the above-referenced application, assigned by its inventors.

**II. Related Appeals And Interferences**

There are no known related appeals.

**III. Status Of Claims**

Claims 1-18 stand finally rejected under 35 U.S.C. §102(e) as allegedly being anticipated by U.S. Patent No. 6,493,717 to Junkin (hereinafter "Junkin"). The final rejection of claims 1-18 is appealed.

**IV. Status Of Amendments**

No amendments to the claims have been filed after the final rejection.

**V. Summary Of Claimed Subject Matter**

A concise explanation of independent claims 1, 7, and 13 is reproduced below, along with a citation to page number, line number, figures, and reference characters, where appropriate, to assist the Board of Patent Appeals and Interferences (hereinafter referred to as “the Board”) in appreciating the significant advances made by the embodiments of the present invention.

**A. Concise explanation of Independent Claim 1**

Claim 1 recites a method for implementing database connection mapping for connecting a user to at least one database in a reporting system (see, for example, FIGs. 1-2 and 8, page 5, ll. 21-22, page 21, line 4 to page 25, line 2), that includes the steps of:

enabling a user to submit a user identification input and a user request to a reporting system (see, for example, FIGs. 3 and 9, page 12, line 19 to page 14, line 15, page 22, ll. 8-10, page 26, ll. 3-16);

identifying the user based on user identification input (see, for example, FIG. 9, page 26, line 3 to page 27, line 22); and

controlling access to at least one database through a centralized server (see, for example, FIG. 8, ‘820,’ page 21, line 15 to page 23, line 2) wherein the centralized server maps the user to at least one appropriate database based on the user request and at least one database connection definition (see, for example, FIGs. 8-10, page 21, line 15 to page 31, line 7).

**B. Concise explanation of Independent Claim 7**

Claim 7 recites a system for implementing database connection mapping for connecting a user to at least one database in a reporting system (see, for example, FIGs. 1-2, 8, page 5, ll. 21-22, page 21, line 4 to page 25, line 2), that includes:

a user input for enabling a user to submit a user identification input and a user request to a reporting system (see, for example, FIGs. 1, 3, 9, and 10, '102,' '1010,' page 12, line 19 to page 14, line 15, page 22, ll. 8-10, page 26, ll. 3-16);

an identification module for identifying the user based on user identification input (see, for example, FIGs. 8-10, page 26, line 3 to page 27, line 22); and

a centralized server for controlling access to at least one database (see, for example, FIG. 8, '820,' page 21, line 15 to page 23, line 2) wherein the centralized server maps the user to at least one appropriate database based on the user request and at least one database connection definition (see, for example, FIGs. 8-10, page 21, line 15 to page 31, line 7).

**C. Concise explanation of Independent Claim 13**

Claim 13 recites a processor-readable medium comprising code for execution by a processor to implement database connection mapping for connecting a user to at least one database in a reporting system (see, for example, FIGs. 1-2, 8, page 5, ll. 21-22, page 21, line 4 to page 25, line 2), the medium includes:

code for causing a processor to enable a user to submit a user identification input and a user request to a reporting system (see, for example, FIGs. 3 and 9, page 12, line 19 to page 14, line 15, page 22, ll. 8-10, page 26, ll. 3-16);

code for causing a processor to identify the user based on user identification input (see, for example, FIG. 9, page 26, line 3 to page 27, line 22); and

code for causing a processor to control access to at least one database through a centralized server (see, for example, FIG. 8, '820,' page 21, line 15 to page 23, line 2) wherein the centralized server maps the user to at least one appropriate database based on the user request and at least one database connection definition (see, for example, FIGs. 8-10, page 21, line 15- page 31, line 7).

## **VI. Grounds Of Rejection To Be Reviewed On Appeal**

The following grounds of rejection are to be reviewed on appeal:

- The rejection of claims 1-18 under 35 U.S.C. §102(e) as allegedly being anticipated by Junkin (U.S. Patent No. 6,493,717).

## **VII. Argument**

The Office has failed to show anticipation of the independent claims. The Office also has failed to establish anticipation of several dependent claims. Each of the specific claims and the impropriety of the rejections is addressed below.

### **A. Rejection of Claims 1-18 Under 35 U.S.C. § 102(e)**

On pages 3-5, the Examiner rejects claims 1-18 under 35 U.S.C. § 102(e) as allegedly being anticipated by Junkin (U.S. Patent No. 6,493,717). Appellants respectfully traverse this rejection.

#### i. Response to the Rejection of Claims 1, 7, and 13

Claim 1 recites:

A method for implementing database connection mapping for connecting a user to at least one database in a reporting system, comprising the steps of:  
enabling a user to submit a user identification input and a user request to a reporting system;  
identifying the user based on user identification input; and

*controlling access* to at least one *database through a centralized server* wherein the *centralized server maps* the *user* to at least one appropriate *database based on* the *user request* and at least one *database connection definition*.  
(Emphasis added.)

Claim 7 recites:

A system for implementing database connection mapping for connecting a user to at least one database in a reporting system, comprising:  
a user input for enabling a user to submit a user identification input and a user request to a reporting system;  
an identification module for identifying the user based on user identification input; and

*a centralized server* for *controlling access* to at least one *database* wherein the *centralized server maps* the user to at least one appropriate database *based on* the *user request* and at least one *database connection definition*.  
(Emphasis added.)

Claim 13 recites:

A processor-readable medium comprising code for execution by a processor to implement database connection mapping for connecting a user to at least one database in a reporting system, the medium comprising:  
code for causing a processor to enable a user to submit a user identification input and a user request to a reporting system;  
code for causing a processor to identify the user based on user identification input; and  
code for causing a processor to control access to at least one database through a *centralized server* wherein the *centralized server maps* the user to at least one appropriate database *based on* the *user request* and at least one *database connection definition*.

(Emphasis added.)

For at least the following reasons, the Examiner does not properly apply 35 U.S.C.

§102(e) to reject claims 1, 7, and 13 in view of Junkin.

First, Junkin does not disclose a centralized server controlling access to a database. To reject these claim features, page 3 of the Action indicates that:

The Data Crawler application server is the centralized server that communicates [sic, communicates] with a database. The user request and at least one database connection definition is based on the HTML format mentioned above and discussed per (Col 9 lines 6-14, Col 9 lines 40- Col 10 line 29)

On page 4, the Action further states that:

Junkin teaches the Datacrawler system controls access to databases from users. All requests of users would be generated to HTML pages. Moreover a Program, which is an application process of the Datacrawler system, determines a database must be accessed by the user based the HTML content) (Col. 9, L. 6-14, L. 40-67 and Col. 10, L. 1-29).

Thus, the Examiner appears to be arguing that Junkin discloses that the Datacrawler application server 14 controls access to external database servers 16a-b. (See also Junkin, FIG. 1.)

Appellants respectfully disagree.

Contrary to the allegations made by the Examiner on pages 3-4 of the Action, Junkin discloses decentralized database access control. Specifically, Junkin discloses that access to the databases is decentrally controlled through database administrators, not through the Datacrawler application server 14, as alleged in the Action. At two different locations, Junkin explicitly states that the database administrators, and not the Datacrawler application server 14, control access to the databases. First, Junkin discloses that “[m]ember access to a Database is controlled exclusively by the Database Administrators.” (See Junkin, col. 28, ll. 34-35; emphasis added.) Second, Junkin discloses that “[d]atabase *access* is controlled entirely by the one or more Database Administrators.” (See Junkin, col. 29, ll. 47-48; emphasis added.) Hence, the decentralized Database Administrators, and not the Datacrawler application server 14, control access to the databases in Junkin. Thus, the Datacrawler application server 14 of Junkin does not anticipate the claimed “centralized server.”

In rejecting claim 1, the Examiner appears to erroneously rely on the authentication procedure disclosed in Junkin where the Datacrawler system authenticates a user as disclosing that the Datacrawler application server 14 of Junkin controls access to a database (see Action, page 4 citing column 28, ll. 54-63 of Junkin). Once authenticated, however, Junkin discloses that the system application interfaces of the Datacrawler system control the “*interaction* between the end-user operator and one or more back-end database(s).” (See Junkin, col. 8, ll. 30-38;



emphasis added.) In other words, Junkin discloses controlling *interaction* with the databases through the Datacrawler system, not controlling *access* to the databases. Junkin discloses that access to a database is “exclusively” and “entirely” controlled by the Database Administrators, not the Datacrawler application server 14 (see Junkin, col. 28, ll. 34-35, col. 29, ll. 47-48.) Moreover, Junkin further discloses in the “Background of the Invention” that a user may use a “database management application program” to interact with a database. (See Junkin, col. 2, ll. 16-34.) In effect, a user may *bypass* the Datacrawler system of Junkin using a “database management application program” to interact with a database. Hence, Junkin discloses that Database Administrators of respective databases decentrally control access to the databases, and Junkin does not disclose that the Datacrawler application server 14 controls access to the databases. Thus, Junkin does not disclose controlling access to databases through a “centralized” Datacrawler application server 14, in contrast with the allegations made in the Action.

Junkin also does not disclose that the Datacrawler system requires a user to be authenticated by the Datacrawler system in order to access the databases. Instead, Junkin discloses that a user of the DataCrawler system may utilize a “guest account” that permits the user to *access* database information without requiring a user to login. Specifically, Junkin discloses that:

All users entering the [Datacrawler] system are assigned to the *guest account* until after a successful login is accomplished. The Guest account is an especially desirable aspect of the DataCrawler system because it allows direct access to database information while ***bypassing the typical login process***. Database Administrators decide how to allow access for this Guest user. By default, databases allow guest access in a read-only mode. (See Junkin, col. 29, ll. 3-8; emphasis added.)

Junkin also discloses that:

The DataCrawler system uses a membership process to allow users entry into a database. A username is linked to a UserGroup from within the database. Whenever the user enters a database, the membership records are checked. If a record is found for that user, the user is allowed access as a member of the defined UserGroup. If a membership is not

found, then the database User Access Configuration determines if and how the user is allowed to enter the Database.  
(See Junkin, col. 30, ll. 11-18; emphasis added.)

Hence, Junkin discloses that access to a database is controlled through a database User Access Configuration setup by a Database Administrator. In other words, Junkin discloses that access to databases is controlled through Database Administrators of *decentralized* databases (See also Junkin, FIG. 1, 16a-b), and not through a centralized database and also not through the Datacrawler application server 14. The Examiner incorrectly asserts that the Datacrawler application server 14 anticipates the claimed centralized server since Junkin does not disclose that the Datacrawler application server 14 controls access to the databases and since Junkin also does not disclose that the Database Administrators centrally control access to the databases. Thus, the Examiner has not properly indicated that Junkin anticipates, under 35 U.S.C. § 102, the claim features of: (1) “controlling access to at least one database through a centralized server.” as recited in claim 1; (2) “a centralized server for controlling access to at least one database,” as recited in claim 7; or (3) “code for causing a processor to control access to at least one database through a centralized server,” as recited in claim 13.

Second, Junkin does not disclose a centralized server that maps a user to at least one appropriate database based on a user request and at least one database connection definition. To reject these claim features recited in claims 1, 7, and 13, the Examiner states on page 4 of the Action that:

Junkin teaches the Datacrawler system controls access to databases from users. All requests of users would be generated to HTML pages. Moreover a Program, which is an application process of the Datacrawler system, determines a database must be accessed by the user based the HTML content) (Col. 9, L. 6-14, L. 40-67 and Col. 10, L. 1-29).

Column 9, lines 6-14 of Junkin state that:

The DataCrawler system incorporates the use of standard Web platform dynamically generated HTML pages for each user request. The Web server is responsible for handling

the incoming and outgoing requests, while the Program application controls the actual HTML content. The Program is the main application process of the DataCrawler system. The Program application requires its own system and configuration storage and interacts with one or more end-user databases.

Column 9, line 40 to column 10, line 29 of Junkin state that:

In operation, the Program executes as follows. The Program receives a URL address along with an argument list of KEY=VALUE pairs from the end-user. The URL tells the Program the client from which the user request originated and the Program name. The Argument list that is generated contains the essential parameters used to generate the actual HTML page returned to the user. The Program then uses the arguments to determine which initial state/configuration/database files must be restored, and uses the information found in the files to open more state/configuration/database files. This process will be repeated many times in the course of processing each user request. While information is retrieved, the system pieces together an HTML page that is formatted for the unique user and the parameters that have returned. The Program then returns the HTML page to the Web server.

Control of the Program is determined by the Argument list. The Argument list provides essentially all of the information needed to identify the user and format the HTML page. This list of Key=Value pairs contains all of the Program variables that are needed by the system.

There are many different arguments used by the system. The most commonly used arguments and the purposes are now described. Symbols may be changed or scrambled, but the function will remain the same.

The "UserID" or "uid" parameter (passed as "u") is a unique Session identifier. The UserID is a string that uniquely identifies the session. It is used by the system to restore the session and state information. The session file is the first file restored by the system, and will include the UserName that identifies the User, as well as other state and session information. Each user is assigned a new unique UserID whenever the user begins a new session and it is retained throughout the duration of the session.

Every session contains a URL. An example of a URL with a uid is as follows:  
crawler?u=393128299954&m=n&db=food\_supply

The "db" parameter (passed as "db") represents the active or requested Database. The Database is a string value that uniquely identifies a Database within the current working directory. If a value is passed, the system will proceed to use this as the working database. All data read and edited will originate from this Database. There can only be one Database being acted upon at a time. An example of a URL with a Database parameter is as follows:  
crawler?u=393128299954&m=n&db=food\_supply

The "source" parameter (passed as "s") uniquely identifies a Node within the active Database. The value is a string that uniquely identifies a Node. When a source value is received, the Program will restore the requested Node Object from the working Database.

The Node Object is responsible for initiating the entire Data view. Only one source is valid at a time.  
(Emphasis added.)

Thus, Junkin discloses a Datacrawler system that receives user requests and a Program that processes the user requests. Junkin also discloses that the Program receives a URL address and an argument list from the end-user. Junkin further discloses that the argument list includes a 'db' parameter that uniquely identifies a database.

In rejecting claims 1, 7, and 13 on pages 3-5 of the Action, the Examiner equates various components from Junkin with various claim features recited in claims 1, 7, and 13. As discussed above, the Examiner equates the claimed centralized server with the Datacrawler application server 14 (see Action, page 3). The Examiner also rejects the claimed user request by citing column 28, lines 54-63 of Junkin, which states that "all users requesting access to a DataCrawler application must first go through the authentication process before gaining entrance to the system." (See Action, pages 3-4.) The Action appears to equate the claimed user request with the "users requests" of Junkin that include a URL address and an argument list. (See Action, page 4 citing col. 9, ll. 40-43 of Junkin.) The Action does not explicitly identify which feature of Junkin allegedly anticipates the claimed database connection definition. The 'db' parameter included in the argument list of Junkin appearing in column 10, lines 12-21 cited on page 4 of the Action will be presumed, *arguendo*, to be the feature of Junkin which the Examiner believes anticipates the claimed database connection definition. Thus, the Action appears to be arguing that Junkin discloses that the Datacrawler application server 14 maps a user to at least one appropriate database based on a "user request" and a 'db' parameter. Appellants respectfully disagree.

Junkin does not disclose that the Datacrawler application server *maps* the user to an appropriate database based on any information. The disclosure of Junkin cited by the Examiner

on pages 3-4 of the Action does not recite the claim term “maps.” The Examiner further does not indicate which component or concept included in columns 9-10 of Junkin that the Examiner believes anticipates the claimed mapping, equivalently, inherently, or otherwise. The Examiner clearly has not met the burden under 35 U.S.C. § 102 of indicating how Junkin discloses that the Datacrawler application server 14 maps the user to a database to anticipate claims 1, 7, and 13.

Nevertheless, Junkin does not disclose that the Datacrawler application server 14 maps the user to at least one appropriate database based on the “user request” and the ‘db’ parameter. Junkin discloses that the “Program *receives* a URL address along with an argument list of KEY=VALUE pairs from the end-user” (see Junkin, col. 9, ll. 41-43; emphasis added) and that the argument list includes a KEY=VALUE pair for a ‘db’ parameter, which includes a string value that uniquely identifies an active or requested database. (See Junkin, col. 9, ll. 57-65, col. 10, ll. 12-22.) Junkin also discloses that:

Every session contains a URL. An example of a URL with a uid is as follows:  
crawler?u=393128299954&m=n&db=food\_supply  
(See Junkin, col. 10, ll. 8-11; emphasis added.)

Thus, Junkin discloses that the end-user provides, to the Datacrawler system, a URL that includes KEY=VALUE pairs for a user identification (i.e., u = 393128299954) and for a database (i.e., db=food\_supply). In other words, Junkin discloses that a user identification provided by the end-user is already associated with a database when the Datacrawler system receives the URL. Hence, Junkin does not disclose that the Datacrawler system associates, maps, or relates in any other manner, the user to a database based on the “user request” and the ‘db’ parameter since the URL provided by the end-user already associates the user identification with the database ‘db’ parameter. Thus, the Examiner has not demonstrated that Junkin anticipates, under 35 U.S.C. § 102, the claim features of: (1) a “centralized server [that] maps the user to at least one appropriate database based on the user request and at least one database

connection definition,” as recited in claim 1; (2) a “centralized server [that] maps the user to at least one appropriate database based on the user request and at least one database connection definition,” as recited in claim 7; or (3) a “centralized server [that] maps the user to at least one appropriate database based on the user request and at least one database connection definition, as recited in claim 13.

Additionally, Junkin uses the term “map” in two contexts, neither of which anticipate the claimed “*centralized server [that] maps the *user* to at least one appropriate *database based on* the *user request* and at least one *database connection definition*” (emphasis added), as recited in claims 1, 7, and 13. The first context using the term “map” in Junkin relates to a “map of hierarchical activity.” Specifically, Junkin discloses that a Universal Data Exander (UDE):*

provides the end-user with a virtual window into the database. The UDE allows the end-user to explore information represented in any number of possible relationships from simple views of a single data source to highly complex views of many different interrelated data sources.

The UDE will display a hierarchical representation of the data by first walking up the hierarchy from each Child Node to its Parent Node by restoring each successive parent. This gives the system a map of the hierarchical activity. Then the system traverses this map and formats the page for the end-user. The process of traversing the “map” of the active hierarchy is a complex process that involves the use of many other objects. (Emphasis added.)

Junkin further depicts a “Parent-Child structure” in FIG. 19. (See also Junkin, col. 13, ll. 42-56.) Clearly, Junkin does not disclose that the hierarchical map of the “Parent-Child structure” depicted in FIG. 19 “maps the *user* to at least one appropriate *database based on* the *user request* and at least one *database connection definition*” (emphasis added), as recited in claim 1.

The second context that Junkin uses the term “map” is discussed in column 31, lines 59-67, which state:

Regarding Data Type mapping, the DataCrawler system will automatically create each SQL query clause for selecting or updating database information. In order to accomplish this, the system needs to map each field's data type into one of three classifications: text, numeric, and dates. DataCrawler will then use this mapping to properly format each *field*

in the *query*. "Text" is enclosed in quotes, "numeric" receives no formatting, and "dates" receives special database dependant formatting.  
(Emphasis added.)

Thus, Junkin discloses that the Datacrawler system maps fields in a query into one of three classifications: text, numeric, and dates. Clearly, Junkin does not disclose that the mapping of query fields into one of three classifications to properly format fields in a query involves mapping "the *user* to at least one appropriate *database based on* the *user request* and at least one *database connection definition*" (emphasis added), as recited in claims 1, 7, and 13. Thus, for at least the reasons provided above, the Examiner has not properly shown that Junkin anticipates claims 1, 7, or 13 under 35 U.S.C. § 102, and claims 1, 7, and 13 are believed to be allowable over Junkin.

Claims 2-6, which depend from claim 1, also are believed to be in condition for allowance due to their dependence on an allowable claim.

Claims 8-12, which depend from claim 7, also are believed to be in condition for allowance due to their dependence on an allowable claim.

Claims 14-18, which depend from claim 13, also are believed to be in condition for allowance due to their dependence on an allowable claim.

Claims 2, 3, 6, 8, 9, 12, 14, 15, and 18 also are separately patentable over Junkin for the reasons discussed below.

ii. Response to the Rejection of Claims 2, 8, and 14

Claim 2 recites:

The method of claim 1 wherein the database connection definition comprises a data source name and a set of properties for establishing a database connection to at least one database.

Claim 8 recites:

The system of claim 7 wherein the database connection definition comprises a data source name and a set of properties for establishing a database connection to at least one database.

Claim 14 recites:

The medium of claim 13 wherein the database connection definition comprises a data source name and a set of properties for establishing a database connection to at least one database.

For at least the following reasons, Junkin does not anticipate claims 2, 8, or 14 under 35 U.S.C. § 102.

On page 4, the Examiner cites column 6, lines 51-56 of Junkin as allegedly anticipating claims 2, 8, and 14. Appellants respectfully disagree. The columns of Junkin cited by the Examiner do not disclose “a data source name” or “a set of properties for establishing a database connection to at least one database.” Instead, column 6, lines 51-56 of Junkin states that:

Data sources and nodes are building blocks of an application supplied by the DataCrawler application server. Each data source defines a portion of the database system, and each node represents a data source. When linked hierarchically, the nodes implement an application in which portions of the database system are linked hierarchically.

This column cited by the Examiner does not use the phrase “data source *name*” nor does this column use the language “a set of properties for establishing a database connection to at least one database.” It is unclear how a discussion of a hierarchical linking of “data sources and nodes” could be construed to anticipate the claim language of a database connection definition that comprises “a data source name and a set of properties for establishing a database connection to at least one database.” Appellants submit that the column of Junkin cited by the Examiner fails to disclose these claim features, and this citation of Junkin certainly does not anticipate claims 2, 8, and 14 under 35 U.S.C. § 102. Thus, the Examiner has not demonstrated that Junkin discloses: (1) “the database connection definition comprises a data source name and a set of properties for



establishing a database connection to at least one database,” as recited in claim 2; (2) “the database connection definition comprises a data source name and a set of properties for establishing a database connection to at least one database,” as recited in claim 8; or (3) “the database connection definition comprises a data source name and a set of properties for establishing a database connection to at least one database,” as recited in claim 14. Therefore, claims 2, 8, and 14 are believed to be separately patentable over Junkin.

iii. Response to the Rejection of Claims 3, 9, and 15

Claim 3 recites:

The method of claim 2 wherein the data source name comprises information for locating and logging into a database.

Claim 9 recites:

The system of claim 8 wherein the data source name comprises information for locating and logging into a database.

Claim 15 recites:

The medium of claim 14 wherein the data source name comprises information for locating and logging into a database.

Junkin does not anticipate claims 3, 9, and 15 under 35 U.S.C. § 102 for at least the following reasons.

Junkin does not disclose a data source name that includes information for locating and logging into a database. To reject these claim features, the Action cites column 23, lines 59-61, column 24, lines 31-34, and column 28, lines 33-40, which respectively recite:

A "DataSource Object" defines the origin of a Data Source, which includes the underlying structure as well as the format of each Field.

A Data Source object defines the type, name and location of each external source of data used within DataCrawler. Options that pertain to each data format also may be specified here.

The second is Database Access. Member access to a Database is controlled exclusively by the Database Administrators. The Administrators control user access via system

configuration and group management. User access refers to how users enter and login to the DataCrawler system. With respect to user accounts, users are global entities within the DataCrawler system.

Appellants note that Junkin also discloses in column 16, lines 21-29 that:

The DataSource Object provides the system with the methods and configuration necessary for retrieving information from a specific source of Structured Information. Within the Display Object, the DataSource Object holds the formatting configuration needed to display a single record of database information. The DataSource also acts as a Parent container for Field Group and Field Objects that compose and detail the actual structure within the DataSource as well as define the display formatting for the End-User.

The Examiner appears to be equating the claimed “data source name” with the “Datasource object” of Junkin. Thus, the Examiner appears to be arguing that the Datasource object includes information for locating and logging into a database. Appellants respectfully disagree.

Junkin does not disclose that the Datasource object includes information for logging into a database. Junkin discloses that the Datasource object “defines the origin of a Data Source” and “defines the type, name and location of each external source of data used within DataCrawler.” (See Junkin, col. 23, ll. 59-61, col. 34, ll. 31-34.) Junkin does not, however, disclose that the Datasource object includes any information for logging into a database. Rather, column 28 cited by the Examiner refers to database access controlled by the Database Administrators. The cited lines of column 28 of Junkin do not discuss the Datasource object, and do not disclose that the ***Datasource object*** includes information for logging into a database. Hence, the columns of Junkin cited by the Examiner do not disclose that the Datasource object includes information for locating and logging into a database. Thus, the Examiner has not properly found anticipation under 35 U.S.C. § 102 since the Examiner has not properly shown that Junkin anticipates: (1) “wherein the data source name comprises information for locating and logging into a database,” as recited in claim 3; (2) “wherein the data source name comprises information for locating and logging into a database, as recited in claim 9; or (3) “wherein the data source name comprises

information for locating and logging into a database,” as recited in claim 15. Therefore, claims 3, 9, and 15 are believed to be separately patentable over Junkin.

iv. Response to the Rejection of Claims 6, 12, and 18

Claim 6 recites:

The method of claim 1 further comprising the step of load balancing query volume associated with the at least one database.

Claim 12 recites:

The system of claim 7 further comprising a load balancing module for load balancing query volume associated with the at least one database.

Claim 18 recites:

The medium of claim 13 further comprising code for causing a processor to load balance query volume associated with the at least one database.

To reject claims 6, 12, and 18, on page 5 of the Action, the Examiner cites column 8, lines 36-38 and lines 45-47 of Junkin, which respectively disclose that:

The End-User interface controls the interaction between the end-user operator and one or more back-end database(s), typically SQL databases.

This operator uses the DataCrawler system to create, deploy and maintain end-user interfaces that will allow access to databases.  
(See Junkin, col. 8, ll. 36-38, 45-47.)

On page 5, the Examiner further states that:

Junkin teaches the system creates SQL queries to be sent to an SQL database. Moreover, there are many SQL databases into the end-user databases. Therefore, it implies to have a number of queries would be created to be sent to those databases.  
(See Action, page 5.)

Thus, Junkin discloses that the Datacrawler system includes an end-user interface that controls end-user interaction with databases. Neither the explanation provided by the Examiner nor the columns of Junkin cited in the Action discuss either load balancing or query volume for any database. In other columns, Junkin discloses that a user may specify a ‘db’ parameter, and if a string value that uniquely identifies a database associated with the ‘db’ parameter is passed, then

“the system will proceed to use this as the working database.” (See Junkin, col. 10, ll. 12-19.) It appears that all of the users of Junkin could specify the same ‘db’ parameter for a specified database without the Datacrawler system considering load balancing of query volume for the specified database. Hence, the columns of Junkin cited by the Examiner do not disclose any of: a step of load balancing; a load balancing module; or a code for causing a processor to load balance. Thus, the Examiner has not properly demonstrated that Junkin anticipates the claim features of: (1) “load balancing query volume associated with the at least one database,” as recited in claim 6; (2) “a load balancing module for load balancing query volume associated with the at least one database,” as recited in claim 12, or (3) “code for causing a processor to load balance query volume associated with the at least one database,” as recited in claim 18. Therefore, claims 6, 12, and 18 are believed to be separately patentable over Junkin.

**VIII. Conclusion**

Because the cited reference fails to disclose all features set forth in the pending claims, Appellants submit that the pending claims are allowable over the cited reference. Accordingly, Appellants respectfully request that the Board reverse the prior art rejections set forth in the Action, and allow all of the pending claims.

Respectfully submitted,

Date: June 23, 2006

By:



Brian M. Buroker  
Registration No. 39,125

Hunton & Williams  
1900 K. St., NW, Suite 1200  
Washington, D.C. 20006-1109  
(202) 955-1894

**IX. Claims Appendix**

1 (Original). A method for implementing database connection mapping for connecting a user to at least one database in a reporting system, comprising the steps of:

enabling a user to submit a user identification input and a user request to a reporting system;

identifying the user based on user identification input; and

controlling access to at least one database through a centralized server wherein the centralized server maps the user to at least one appropriate database based on the user request and at least one database connection definition.

2 (Original). The method of claim 1 wherein the database connection definition comprises a data source name and a set of properties for establishing a database connection to at least one database.

3 (Original). The method of claim 2 wherein the data source name comprises information for locating and logging into a database.

4 (Original). The method of claim 2 wherein the database connection comprises a physical open database connectivity connection to a database.

5 (Original). The method of claim 1 wherein the user is associated with a group of users where each user of the group is mapped to a database connection via a database login.

6 (Original). The method of claim 1 further comprising the step of load balancing query volume associated with the at least one database.

7 (Original). A system for implementing database connection mapping for connecting a user to at least one database in a reporting system, comprising:

a user input for enabling a user to submit a user identification input and a user request to a reporting system;

an identification module for identifying the user based on user identification input; and

a centralized server for controlling access to at least one database wherein the centralized server maps the user to at least one appropriate database based on the user request and at least one database connection definition.

8 (Original). The system of claim 7 wherein the database connection definition comprises a data source name and a set of properties for establishing a database connection to at least one database.

9 (Original). The system of claim 8 wherein the data source name comprises information for locating and logging into a database.

10 (Original). The system of claim 8 wherein the database connection comprises a physical open database connectivity connection to a database.

11 (Original). The system of claim 7 wherein the user is associated with a group of users where each user of the group is mapped to a database connection via a database login.

12 (Original). The system of claim 7 further comprising a load balancing module for load balancing query volume associated with the at least one database.

13 (Original). A processor-readable medium comprising code for execution by a processor to implement database connection mapping for connecting a user to at least one database in a reporting system, the medium comprising:

code for causing a processor to enable a user to submit a user identification input and a user request to a reporting system;

code for causing a processor to identify the user based on user identification input; and

code for causing a processor to control access to at least one database through a centralized server wherein the centralized server maps the user to at least one appropriate database based on the user request and at least one database connection definition.

14 (Original). The medium of claim 13 wherein the database connection definition comprises a data source name and a set of properties for establishing a database connection to at least one database.

15 (Original). The medium of claim 14 wherein the data source name comprises information for locating and logging into a database.

16 (Original). The medium of claim 14 wherein the database connection comprises a physical open database connectivity connection to a database.

17 (Original). The medium of claim 13 wherein the user is associated with a group of users where each user of the group is mapped to a database connection via a database login.

18 (Original). The medium of claim 13 further comprising code for causing a processor to load balance query volume associated with the at least one database.



**X. Evidence Appendix**

None.

**XI. Related Proceedings Appendix**

None.